IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

FILING OF A UNITED STATES PATENT APPLICATION

METHOD AND APPARATUS FOR INTERPOLATING PIXEL PARAMETERS BASED ON A PLURALITY OF VERTEX VALUES

INVENTORS:

Andrew Gruber
215 Pleasant Street
Arlington, MA 02476

ATTORNEY OF RECORD:
TIMOTHY J. BECHEN
REGISTRATION NO. 48,126
VEDDER, PRICE, KAUFMAN & KAMMHOLZ
222 NORTH LASALLE STREET, SUITE 2600
CHICAGO, ILLINOIS 60601
PHONE (312) 609-7500
FAX (312) 609-5005

Express Mail Label No.: EV 320528262 US

Date of Deposit: 08/01/03
I hereby certify that this paper is being deposited
with the U.S. Postal Service "Express Mail Post
Office to Addressee" service under 37 C.F.R.
Section 1.10 on the date of deposit, indicated
above, and is addressed to: Assistant
Commissioner for Patents, Box Patent Application,
Washington, DC 20231.

Name of Depositor: Karenina Oliver
Signature:

METHOD AND APPARATUS FOR INTERPOLATING PIXEL PARAMETERS BASED ON A PLURALITY OF VERTEX VALUES

FIELD OF THE INVENTION

[0001] The present invention relates generally to graphics processing and more specifically to parameter interpolation in a three-dimensional graphics processing system.

BACKGROUND OF THE INVENTION

[0002] In a typical graphics processing system, one common graphics processing technique is the generation of images through the composition of a plurality of vertices. The plurality of vertices define a specific region containing a plurality of pixels therein. In common hardware implementations, interpolation of a pixel parameter is based on slopes of the value of the parameters at the vertices, for example, the value of a material color or texture at each of the three vertices for a triangle. Typically, the values are assumed to be planar wherein the three points of the triangle, vertices, provide all the requisite information for defining any parameter value over the entire x,y plane.

[0003] Typically, this interpolation process is implemented in two steps. The first step, a setup mode, is a once per triangle step of providing the slope of the parameter relative to x and y coordinates of a display screen. The slope may be denoted as a change in the parameter value divided by a change in an x plane and a change in the parameter value divided by a difference in the y plane. Furthermore, for each pixel within the triangle, the value at that point is calculated using these two slopes. The starting vertex is denoted as P_0 and is assumed to be located at (x_0, y_0). For illustration purposes only, the other two vertices are denoted as P1, located at (x_1, y_1), and P2 is located at (x_2, y_2). Therefore, using equation 1, a parameter value for a particular location may be determined.

[0004] Equation #1: $P(x,y) = P0 + (x-x_0)*dP/dx + (y-y_0)*dP/dy$

2

[0005] A problem occurs in a typical processing system because of processing requirements to maintain an adequate level of precision. In a typical processing system, it is possible for the slopes in the dx and dy direction to have very large magnitudes, but opposite signs. The two slopes may largely cancel each other out, thereby leaving a reasonable value only if enough precision is used for the slope terms. For example, if a four bit sub-pixel precision is used in a screen having a 2Kx2K bit precision, the parameter whose range within the triangle is limited to (0,.1) may have a slope as much as 16 providing for the range of parameters across a screen to be as high as 32,768. Therefore, to accurately represent such a parameter to 24 bits of precision below the decimal point, the slope must be represented to a precision of 39 bits. Such precision is very costly to calculate in a setup engine, as well as costly to use for per-pixel interpolation. Not only are there transactional costs in terms processing cycles for computing values at the desired precision level, but there also exists costs in processing speed for determining these parameters.

[0006] Processing costs exists for parameter calculations across both the setup mode and a calculation mode of operation. There exists a need for a pixel processing method and apparatus to allow for parameter interpolation wherein the calculation of terms within the setup mode and calculations during the calculation mode provide improved precision and efficiency through a reduction in overall calculations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1 illustrates a graphical representation of a graphical triangle having a plurality of vertices and are a plurality of pixel parameters;

3

[0008] FIG. 2 illustrates a flowchart of a method for interpolating pixel parameters based on a plurality of vertex values in accordance with one embodiment of the present invention;

[0009] FIG. 3 illustrates a flowchart of a calculation mode of one embodiment of a method for interpolating pixel parameters based on the plurality of vertex values;

[0010] FIG. 4 illustrates a flowchart of an alternative method for interpolating pixel parameters based on the plurality of vertex values;

[0011] FIG. 5 illustrates an apparatus for interpolating pixel parameters based on the plurality of vertex values, in accordance with one embodiment of the present invention; and

[0012] FIG. 6 illustrates computing system utilizing the graphics processing unit of FIG. 5, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[0013] Generally, a method and apparatus for interpolating pixel parameters based on a plurality of vertex values includes operating first in a setup mode and then in a calculation mode. The method and apparatus includes, while in a setup mode, generating a plurality of differential geometric values based on the plurality of vertex values, wherein the differential geometric values are independent of a parameter slope between the pluralities of vertex values. The typical vertex value includes a general value having a (x, y) coordinate value and a differential geometric value, which is a value computed based on a difference of specific vertex values.

[0014] The method and apparatus further includes, while in a calculation mode, determining a first geometric value and a second geometric value based on a pixel value, the plurality of vertex values and the differential geometric values. The first geometric value and second geometric value are values calculated based on combination of mathematical manipulation of various pixel values and differential geometric values. Furthermore, a pixel value is determined for each of the plurality of pixels based on the vertex parameter value, the first geometric value and the second geometric value. In the preferred embodiment, the first geometric value, second geometric value and the pixel value are determined for each of the plurality of pixels, on a pixel by pixel basis.

[0015] In one embodiment, the plurality of differential geometric values may be stored in a temporary buffer during the setup mode and read from the temporary buffer during the calculation mode. The temporary buffer may be any type of memory device, but not limited to, single memory, plurality of memory locations, shared memory, CD, DVD, ROM, RAM, EEPROM, optical storage or any other non-volatile storage medium capable of storing differential geometric values therein.

[0016] More specifically, FIG. 1 illustrates a graphical representation of a triangle 100 having three (3) vertices 102, 104 and 106. Each of the vertices 102, 104 and 106 includes a first_plane value (x) and a second_plane value (y). More specifically, the zero_vertex parameter 102 includes the first_plane zero_vertex parameter (x0) and second_plane zero_vertex parameter (y0). The parameter 104 includes the first_plane first_vertex parameter (x1) and the second_plane first_vertex parameter (y1). The second_vertex parameter 106 includes a first_plane second_vertex parameter (x2) and a second_plane second_vertex parameter (y2).

5

[0017] When the vertices 102, 104 and 106 define a triangle, a pixel parameter 108 may be calculated based on the interpolation of the various vertex parameters, as discussed below.

[0018] FIG. 2 illustrates a flowchart representing the steps of a method for interpolating the pixel parameters 108 based on the plurality of vertex values 102, 104 and 106. The method begins, step 200, by operating first within a setup mode. The first step, step 202, while in the setup mode, is generating a plurality of differential geometric values based on a plurality of vertex values, wherein the differential geometric values are independent of a parameter slope between the plurality of vertex values. The differential geometric values are differences between specific plane vertex values. In one embodiment, the plurality of differential geometric values may be calculated based on the following equations:

[0019] Equation #2: $dx02 = x0\text{-}x2$

[0020] Equation #3: $dy02 = y0\text{-}y2$

[0021] Equation #4: $dx10 = x1\text{-}x0$

[0022] Equation #5: $dy10 = y1\text{-}y0$

[0023] In one embodiment, the first differential geometric value is calculated based on the difference between the first_plane zero_vertex parameter (102) and the first_plane second_vertex parameter (106). The second differential geometric value is calculated based on the difference between the second_plane zero_vertex parameter (102) and the second_plane second_vertex parameter (106). The third differential geometric value is

6

calculated based on the difference between the first_plane first_vertex parameter (104) and the first_plane zero_vertex parameter (102). And the fourth differential geometric value is calculated based on the difference between the second_plane first_vertex parameter (104) and the second_plane zero_vertex parameter (102).

[0024] In one embodiment of the present invention, the method next operates in a calculation mode wherein the next step, step 204, is determining a first geometric value and a second geometric value based on a pixel value, the plurality of vertex values and the differential geometric values. In one embodiment, the first geometric value may be calculated by equation 6 and the second geometric value may be calculated by equation 7.

[0025] Equation #6: $geom\_1 = (x-x0)*dy02 + (y-y0)*dx02$

[0026] Equation #7: $geom\_2 = (x-xo)*dy10 + (y-y0)*dx10$

[0027] In this embodiment, the first geometric value is calculated by the difference of the first_plane pixel parameter 108 (x) and the first_plane zero_vertex parameter 102 (x0) times the first differential geometric value added to the difference between the second_plane pixel parameter 108 (y) and second_plane zero_vertex parameter 102 (y0) times the first differential geometric value. Furthermore, in this embodiment, the second geometric value is calculated based on the multiplication of the difference between the first_plane pixel parameter 108 (x) and the first_plane zero_pixel parameter 102 (x0) and the fourth differential geometric value summed with the multiplication of the difference between the second_plane pixel parameter (108) and the second_plane zero_vertex parameter 102 (y0) and the third differential geometric value.

7

[0028] Furthermore, while on the calculation mode, the next step, step 206, is determining a pixel value based on a vertex parameter value, the first geometric value and the second geometric value. The pixel value 108, in one embodiment, may be calculated based on the following equation:

[0029] Equation #8: $P(x,y) = P0 + Pdif1*geom\_1 + Pdif2*geom\_2$

[0030] The pixel value is the combination of the vertex parameter (p0), the first geometric value (geom_1) multiplied by a first parameter difference value (Pdif1) and the product of the second geometric value (geom_2) and a second differential parameter value (Pdif2). As discussed in greater detail below, the pixel values calculated by equation 8 include terms, Pdif1 and Pdif2, which are set-up mode terms. These set-up mode terms are calculated during the set-up mode and may be used to verify the value of the pixel value P(x,y), as discussed below utilizing, in one embodiment, equations 9-11.

[0031] As indicated in FIG. 2, steps 204 and 206 are performed while operating in a calculation mode and are performed on a pixel-by-pixel basis, as illustrated by the determinative step 208. In the event there are more pixels for pixel parameters to be calculated, steps 204 and 206 are repeated for each value until all of the pixel parameters are calculated. Thereupon, the method is complete, step 210.

[0032] The method of FIG. 2 improves over prior interpolating pixel parameter techniques by, during the calculation mode, utilizing terms calculated during the setup mode, wherein calculation mode calculations provide for fewer computations and improved precision, thereby improving the processing speed and reducing overhead processing requirements. More specifically, the differential geometric values are

8

calculated in the setup mode and may be reused for each pixel parameter calculation which is done on a pixel-by-pixel basis during the calculation mode. Furthermore, based on calculating Equations 2-5 in setup mode, an optimized level of precision may be obtained without requiring excess computational overhead during the calculations of the calculation mode, as discussed below.

[0033] As recognized by one having ordinary skill in the art, the plurality of differential geometric values and the geometric values may be calculated using any variant combination to produce equivalent results. Regardless thereof, in accordance with one embodiment of the present invention, the geometric differential values may be calculated during the setup mode and the geometric values calculated during the calculation mode on a per pixel basis using differential geometric values.

[0034] FIG. 3 illustrates another embodiment of the calculation mode operation of the method for interpolating pixel parameters based on a plurality of vertex values, wherein a plurality of the terms utilized in the calculation mode are pre-calculated during the set-up mode, as noted below. The method begins, step 300, by generating a first geometric value and a second geometric value for the set of vertex values based on a pixel value, a first vertex value, a second vertex value and a third vertex value, wherein the geometric values are independent of a slope between the plurality of vertex values, step 302. Similar to the embodiment described above, the first geometric value may be calculated using equation 6 and the second geometric value may be calculating using equation 7.

[0035] In a further embodiment, step 302 consists of steps 304 and 306, wherein step 304 includes determining the sum of the product of a first difference and a first differential

9

geometric value and the product of a second difference and a second differential value and step 306 includes determining the sum of the product of the first difference and a third differential value and the product of a third difference and the fourth differential geometric value. In this embodiment, the first difference is the difference between the first_pixel parameter (x) and the first_plane zero_vertex parameter (x0). In one embodiment, the first differential geometric value is the value calculated using equation 2. The second difference is the difference between the second_plane pixel value (y) and the second_plane zero_vertex parameter (y0). Furthermore, in one embodiment, the second differential geometric value is the value calculated by equation 3. In one embodiment, the third differential geometric value is the value calculated by equation 5 and the fourth differential value is the value calculated by equation 4.

[0036] Upon the generation of the first geometric value and the second geometric value, the next step, step 308, is determining a pixel value for each of the plurality of pixels based on the first geometric value adjusted by a first differential value and the second geometric value adjusted by a second geometric value such that the pixel value is determined based on a product of the first differential value and the first geometric value combined with the product of the second differential value and the second geometric value combined with a zero vertex parameter value.

[0037] As discussed above, step 308 is illustrated the equation 8. Step 308 further includes steps 310 and 312. Step 310 includes the first differential value being the difference between a first vertex parameter value and a zero vertex parameter value, the difference divided by a twice_area value and the second differential value is the difference between a second vertex parameter value and the zero vertex parameter value,

10

the difference divided by a twice_area value. In one embodiment, the first differential value and second differential value are calculated during the set-up mode using the following equations:

[0038] Equation #9: Pdif1 = (P1-P0)/twice_area

[0039] Equation #10: Pdif2 = (P2-P0)/twice_area

[0040] Step 312 includes the twice_area value as being a difference between a product of a difference between the first_plane second_vertex parameter and a first_plane zero_vertex parameter and a difference between the second_plane first_vertex parameter and the second_plane zero vertex parameter and a product of a difference between the first_plane zero_vertex parameter and the first_plane first_vertex parameter and a difference between the second_plane zero_vertex parameter and a second_plane second_vertex parameter. As discussed above with regards to equations 8-10, the Pdif terms are terms calculated during the set-up mode, therefore the twice_area term of equation 11 is also calculated during the set-up mode. Stated as an equation, step 312 may be quantified, in one embodiment, as equation 11:

[0041] Equation #11: twice_area = (x2-x0)(y1-y0) – (x0-x1)(y0-y2)

[0042] Thereupon, upon the completion of step 308, while in a calculation mode, the method is complete, step 314. As discussed above, steps 302 and 308 may be repeated on a per-pixel basis utilizing the plurality of differential geometric values repeatedly for each pixel, thereby reducing system processing overhead. Through the utilization of values computed during the set-up mode, the calculation mode efficiently generates the

11

intended values. Furthermore, through generating the values noted in equations 9-11, the calculated value of equation 8 may be verified to insure that the proper pixel value P(x,y) is calculated using the values of equations 9-11.

[0043] FIG. 4 illustrates an alternative embodiment of a method for interpolating pixel parameters. The method begins, step 400, by receiving a zero vertex value, step 402. In one embodiment, zero vertex value may be the value P0(x0, y0). The next step, step 404, is receiving a first vertex value, such as P1(x1, y1). The next step, step 406, is receiving a second vertex value, such as P2(x2, y2).

[0044] Upon receiving the zero vertex value, the first vertex value and the second vertex value, the next step, step 408, is generating a first differential geometric value and a second differential geometric value in relation to the zero vertex value and the second vertex value. Moreover, the first differential geometric value and the second differential geometric value are both independent of a parameter slope between the zero vertex value and the second vertex value. As discussed above, the first differential geometric value is similar to the value generated by equation 2 and the second differential geometric value is the value created by equation 3.

[0045] The next step, step 410, is generating a third differential geometric value and a fourth differential geometric value in relation to the first vertex value and the second vertex value. Once again, the third differential geometric value and the fourth differential geometric value are both independent of a parameter slope between the first vertex value and the second vertex value. As discussed above, the third differential geometric value,

12

in one embodiment, is the value generated by equation 4 and the fourth differential geometric value is the value generated by equation 5.

[0046] Thereupon, the first, second, third and fourth differential geometric values are written to a temporary buffer, step 412. In one embodiment, steps 402-412 are performed during a setup mode of operation.

[0047] Whereupon, during a calculation mode of operation, the first, second, third and fourth differential geometric values are read from the temporary buffer, step 414. Thereupon, on a pixel-by-pixel basis, a pixel value may be determined for each of a plurality of pixels based on a vertex parameter value, a first geometric value and a second geometric value, wherein the first geometric value and the second geometric value are determined based on the first, second, third and fourth differential geometric values, step 416. As discussed above, the first geometric value, in one embodiment, is represented by equation 6 and the second geometric value, in one embodiment, is generated by equation 7.

[0048] Next, step 418, a determination is made as to whether there are more pixels. In the event there are more pixels, step 416 is repeated for each of the pixels of the plurality of pixels. In the event no more pixels are to have parameters interpolated therein, the method is complete, step 420.

[0049] FIG. 5 illustrates an apparatus 500 for interpolating pixel parameters based on a plurality of vertex values, in accordance with one embodiment to the present invention. The apparatus 500 includes a memory 502 coupled to a graphics processing unit 504, wherein the graphics processing unit 504 is coupled to a frame buffer 506. The memory

13

502 may be, but not limited to, a single memory, a plurality of memory locations, shared memory, CD, DVD, ROM, RAM, EEPROM, optical storage, or any other non-volatile storage medium capable of storing digital data thereon. The graphics processing unit 504 further includes a processor 508 coupled to a temporary buffer 510 and a shader 512.

[0050] The processor 508 may be, but not limited to, a single processor, plurality of processors, a DSP, microprocessor, an ASIC, a state machine, or any other implementation capable of processing and executing software or discrete logic or any other suitable combination of hardware, software and/or firmware. The term processor should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include DSP hardware, ROM for storing software, RAM, and any other volatile or non-volatile storage medium. The shader 512 may be a typical graphics shader operating in accordance with known graphics shading techniques to provide shading for a plurality of pixels. Furthermore, as recognized by one having ordinary skills in the art, the temporary buffer 510 may be any type of suitable memory, such as described above with respect to the memory 502.

[0051] The processor 508 is coupled to the temporary buffer 510 across a bus 514 for writing and reading temporarily stored information to and from the temporary buffer 510. In one embodiment, the processor 508 receives a plurality of vertex values 516. In one embodiment, the vertex values represent values P0(x0,y0), P1(x1,y1) and P2(x2,y2). The processor 508 also receives executable instructions 518 from the memory 502 for performing operations therein. The processor 508, in response to the executable instructions 518, operates in one of two different modes, a setup mode and a calculation mode. As discussed above with regard to FIGS. 2-4, while on a setup mode, a plurality

14

of geometric differential values are calculated based on the vertex values and stored in the temporary buffer 510. Whereupon, during a calculation mode, the differential geometric values 520 are read from the temporary buffer 510, across bus 514, such that the processor may perform pixel parameter interpolation on a pixel-by-pixel basis using the differential geometric values.

[0052] Upon generating the interpolated pixel parameters, the processor 508 provides pixel information 522 to the shader 512, in accordance with known shading operations. The shader 512 thereupon performs shading operations, writing a graphical output 524 to the frame buffer 506. Whereupon, the apparatus 500, through the operations of the processor 508 in response to the executable instructions 518 to the memory 502, provides for interpolating pixel parameters based on the plurality of vertex values 516 in the setup and calculation mode system, thereby improving processing speed, reducing overall system processing, and improving parameter precision overhead through the use of pre-generated geometric differential values 520 temporarily stored in the temporary buffer 510.

[0053] FIG. 6 illustrates a visual display system 600 having a computer processing system 602 coupled to a display 604. The display 604 may be any suitable display, such as a computer monitor, LCD screen, a CRT screen, or any other visual output device as recognized by having ordinary skill in the art. The computer processing system 602 includes a CPU 605 operably coupled to system memory 606 via a main bus 608, which is further coupled to the graphics rendering engine 500. As recognized by one having ordinary skill in the art, many elements within the computer processing system 602 and the graphics rendering engine 500 have been omitted for clarity purposes only.

15

[0054] Similar to operations discussed above with regard to FIG. 5 and performing operations as discussed in FIGS. 2-4, in response to executable instructions 518 from the memory 502, the graphics processing unit 504 performs setup modes operations based on vertex values and calculation mode operations based on the plurality of differential geometric values stored in the temporary buffer 510. The graphics processing unit 504 generates the visual display output 524 provided to the frame buffer 506. In accordance with known frame buffer technology, once the frame buffer 506 includes a full screen, a display output 610 is provided to display 604 for visual display thereon.

[0055] The present invention improves over prior art parameter interpolation technique by reducing the amount of system processing required to interpolate specific pixel parameters. Among other benefits, through utilizing setup mode and calculation mode operation, overall system processing overhead is reduced through initial calculations of the differential geometric values (such as equations 2-5). By storing these differential geometric values in a temporary buffer, the specific differential geometric values may be reused on a per-pixel basis to generate geometric values (equations 6 and 7), which is further used to generate pixel parameters (equation 8).

[0056] The present invention further provides for limiting any loss of value precision in the calculated pixel value $P(x,y)$ of equation 8. None of the terms of equation 8, for points within the triangle, exceed the range of the parameter vertex values. As recognized by one having ordinary skill in the art, multiplication and division operations are precision preserving operations, whereas precision may be lost during an addition operation having two large addends with opposite signs. As illustrated by exemplary equation:

16

[0057] Equation #12 : $P(x,y) = P0 + (P1-P0)*i + (P2-P0)*j$

[0058] Within equation 12, the value "i" is calculated by the division of geom._1 (equation 6) by twice_area (equation 11) and the value "j" is calculated by the division of geom._2 (equation 7) by twice_area (equation 11). As noted above, the values "i" and "j" must be less than 1 for all points within the triangle and because the value $P(x,y)$ is an interpolation function, all values of $P(x,y)$ fall within the range of values defined by the vertices.

[0059] If either "i" or "j" exceeds a value of 1, the $P(x,y)$ value would exceed the range of the parameter values at the vertices as $P(x,y)$ would be greater than P1, P2 or P3. This would therein violate the interpolation constraint. As such, because of this limitation on the range of (i,j), the range of each of addend terms of equation 8 are limited. Thereupon, this limits any loss of precision, allowing $P(x,y)$ to be generated at a given precision without the need for greatly extended precision during intermediate calculations.

[0060] It should be understood that there exists implementations of other variations and modifications of the invention and its various aspects, as may be readily apparent to those of ordinary skill in the art, and that the invention is not limited by the specific embodiments described herein. For example, the parameter interpolation method and apparatus may further apply to barycentric coordinates or any other coordinate system, as recognized by one having ordinary skill in the art. It is therefore contemplated and covered by the present invention any and all modifications, variations, or equivalents that fall within the scope of the basic underlying principles disclosed and claimed herein.